Quelques contributions de la Logique Linéaire à la Théorie de la complexité

Séminaire du département « Méthodes Formelles » du Loria.

Clément Aubert

Équipe *Logique De la Programmation*
Institut de Mathématiques de Luminy

19 mars 2014

La Logique Linéaire, c'est. . .

- Franco-français
- Compliqué
- Inutile

La Logique Linéaire, c'est...

Franco-français

Des labos de Mathématiques, d'Informatique et de Philosophie

- en France : Paris $\times 2$, Lyon, Marseille,...
- au Japon (Kyoto), Italie (Bologne, Rome, Turin), Angleterre (Birmingham), Canada (Ottawa), ...

Compliqué

Jeune (GIRARD « Linear logic » 1987), riche et complexe.
Outils audacieux.

Inutile

Cet exposé !

- la décomposition de la logique classique

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'} \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$$

- la décomposition de la logique classique

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'} \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \,\&\, B, \Delta}$$

La Logique Linéaire, c'est...

- la décomposition de la logique classique

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'} \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \,\&\, B, \Delta}$$

- l'attention portée aux ressources

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A} \qquad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}$$

- la décomposition de la logique classique

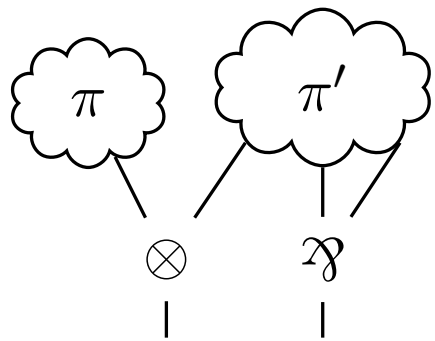
$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'} \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \,\&\, B, \Delta}$$

- l'attention portée aux ressources

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A} \qquad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}$$

- les réseaux de preuves

$$\dfrac{\dfrac{\vdots\ \pi}{\vdash A} \qquad \dfrac{\dfrac{\vdots\ \pi'}{\vdash B, C, D}}{\vdash B, C \parr D}\ \parr}{\vdash A \otimes B, C \parr D}\ \otimes \qquad \dfrac{\dfrac{\vdots\ \pi}{\vdash A} \qquad \dfrac{\vdots\ \pi'}{\vdash B, C, D}}{\vdash A \otimes B, C, D}\ \otimes}{\vdash A \otimes B, C \parr D}\ \parr$$



↬ Un outil ↫

pour l'étude de la

- programmation Quantique
- Modularité
- Concurrence
- Locativité
- Synchronisation (Partage, Parallélisme)
- Complexité !

Objectifs

- Caractériser les classes de complexité indépendamment des modèles de calcul.
- Analyser ou certifier les ressources nécessaires à l'exécution d'un programme.

Techniques

- Récursion borné
- Théorie des modèles
- Quasi-interprétation
- Logique Linéaire

Circuits booléens	**AC**	« Efficace en temps »
	∩	
Temps polynomial	**P**	« Raisonnable »
	∪	
Espace logarithmique	(**N**)**L**	« Économe en espace »

WADLER « Is there a use for linear logic ? » 1991

« Past attempts to apply Girard's linear logic have either had a clear relation to the theory (. . .) or a clear practical value (. . .), but not both. »

WADLER « Is there a use for linear logic ? » 1991

« Past attempts to apply Girard's linear logic have either had a clear relation to the theory (. . .) or a clear practical value (. . .), but not both. »

Un exemple ?

GIRARD « Light Linear Logic » 1998

⇓

BAILLOT et TERUI « Light types for polynomial time computation in lambda-calculus » 2004

⇓

DAL LAGO et SCHÖPP « Type Inference for Sublinear Space Functional Programming » 2010

⇓

`https://github.com/uelis/intml` (2013)

Première partie

Approches syntaxiques

Techniques

Isoler un sous-sytème de la Logique Linéaire

- assez expressif pour encoder des Machines de Turing à complexité bornée
- dont l'élimination des coupures soit à complexité bornée

Résultats

Charactérisation de (**N**)**P**, temps élémentaire, **PSPACE**, **L**

Outils

- Réseaux de preuves
- Bornes* sur $\forall$ et !
- *forks* gérés par les additifs

* : Liens avec la stratification (*tiering*)

(Et la complexité de la Logique Linéaire ?)

- Deux preuves ont-elles la même forme normale ?

P-complet pour MLL
co-NP-complet pour MALL

- Un réseau de preuve est-il correct ?

NL-complet peu importe le fragment

- Une formule est-elle prouvable ?

NP-complet pour MLL.
PSPACE-complet pour MALL
Indécidable pour LL
Ouvert pour MELL

Les réseaux de preuves

- sont des objets parallèles
- à taille d'entrée fixée
- peuvent représenter les fonctions booléennes
- et gérer la duplication tout en étant linéaires.

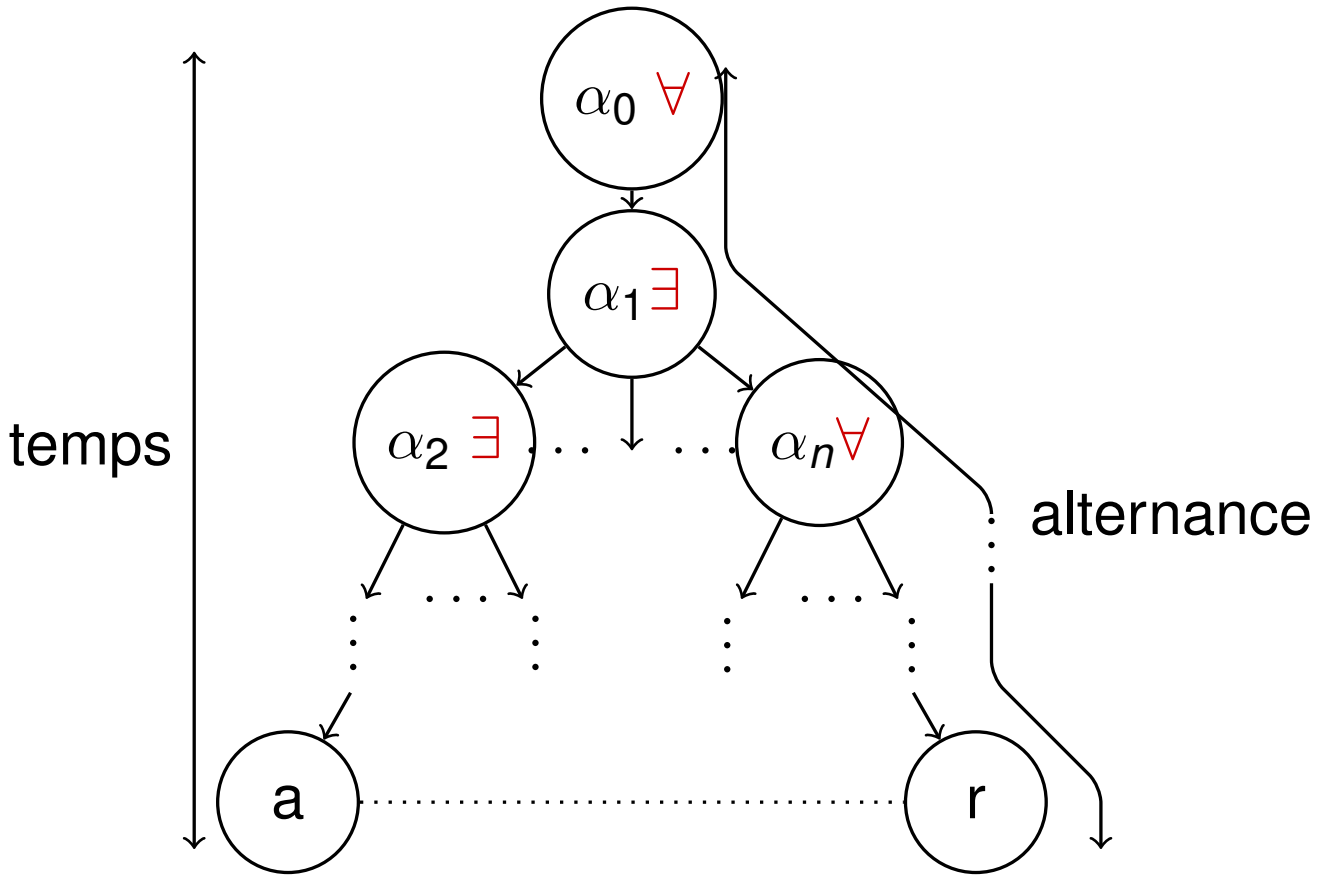
De nouvelles correspondances

- avec les circuits booléens
- avec les machines de Turing alternantes
- qui fonctionnent avec $\mathbf{AC}^0$

Définition

Une machine de Turing alternante à k bandes est

- Q un ensemble fini d'états, $Q = Q_\forall \cup Q_\exists$, $s \in Q$ l'état initial,
- Δ une relation entre $Q \times \Sigma^{k+1}$ et $Q \times \Sigma^k \times \{\leftarrow, \rightarrow, -\}^{k+1}$



Définition ($\mathbf{AC^i}$)

Pour tout $i \in \mathbb{N}$, l'ensemble des fonctions booléennes calculables par une famille uniforme de circuits booléens $C = (C_n)$ où pour tout C_n

- sa profondeur est $\mathcal{O}(\log^i(n))$,
- sa taille est $n^{\mathcal{O}(1)}$,
- avec des portes $\neg$, $\vee^k$ et $\wedge^k$ pour $k \in \mathbb{N}$

Le cas borné $\vee^2$ et $\wedge^2$ définit $\mathbf{NC^i}$

$$\mathbf{AC^i} \subseteq \mathbf{NC^{i+1}} \subseteq \mathbf{AC^{i+1}}$$

Théorème

$$\mathbf{AC^i} = \mathbf{STA}(\log, *, \log^{\mathbf{i}})$$

$$\mathbf{NC^i} = \mathbf{STA}(\log, \log^{\mathbf{i}}, *)$$

Définition ($\mathbf{PCC^i}$)

Pour tout $i \in \mathbb{N}$, l'ensemble des fonctions booléennes calculables par une famille uniforme de circuits de preuves $P = (P_n)$ où pour tout P_n

- sa profondeur est $\mathcal{O}(\log^i(n))$,
- sa taille est $n^{\mathcal{O}(1)}$,
- avec des pieces $\mathrm{Neg}, \mathrm{Dupl}^k, \mathrm{Disj}^k, \mathrm{Conj}^k$ pour $k \in \mathbb{N}$.

Le cas borné Disj^2 et Conj^2 définit $\mathbf{bPCC^i}$

$$\mathbf{PCC^i} \subseteq \mathbf{bPCC^{i+1}} \subseteq \mathbf{PCC^{i+1}}$$

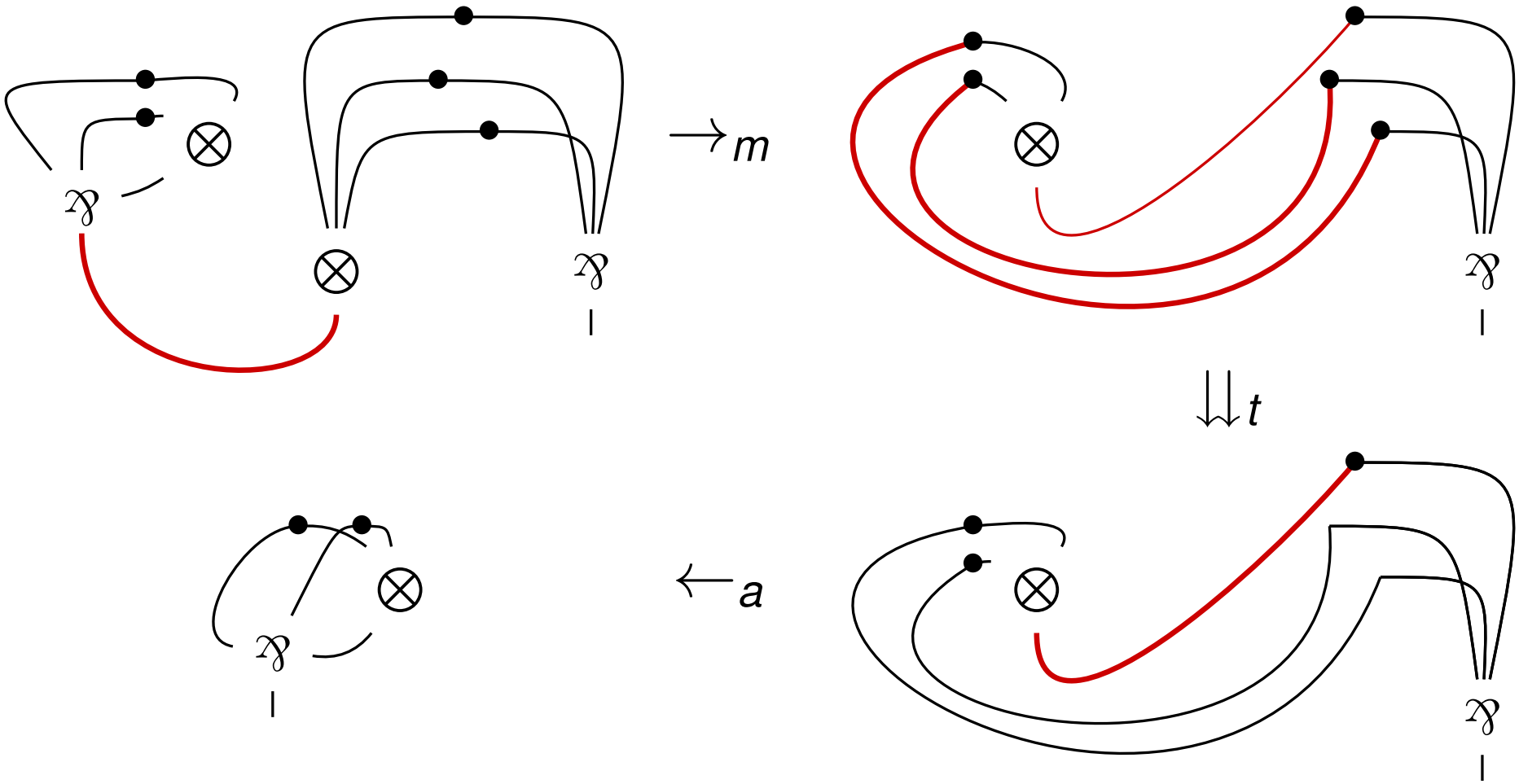
Théorème

$$\mathbf{AC^i} = \mathbf{STA}(\log, *, \log^{\mathbf{i}})$$

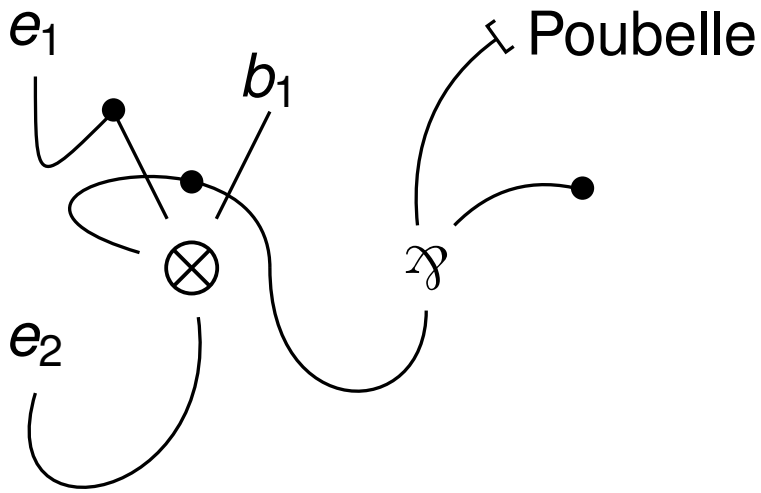
$$\mathbf{NC^i} = \mathbf{STA}(\log, \log^{\mathbf{i}}, *)$$

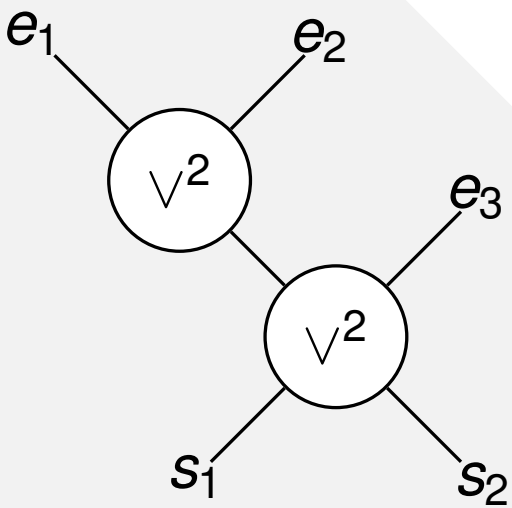
Circuits de preuves : Comment calculer ?

Type booléen : $\parr^3(\alpha^\perp, \alpha^\perp, \otimes^2(\alpha, \alpha))$

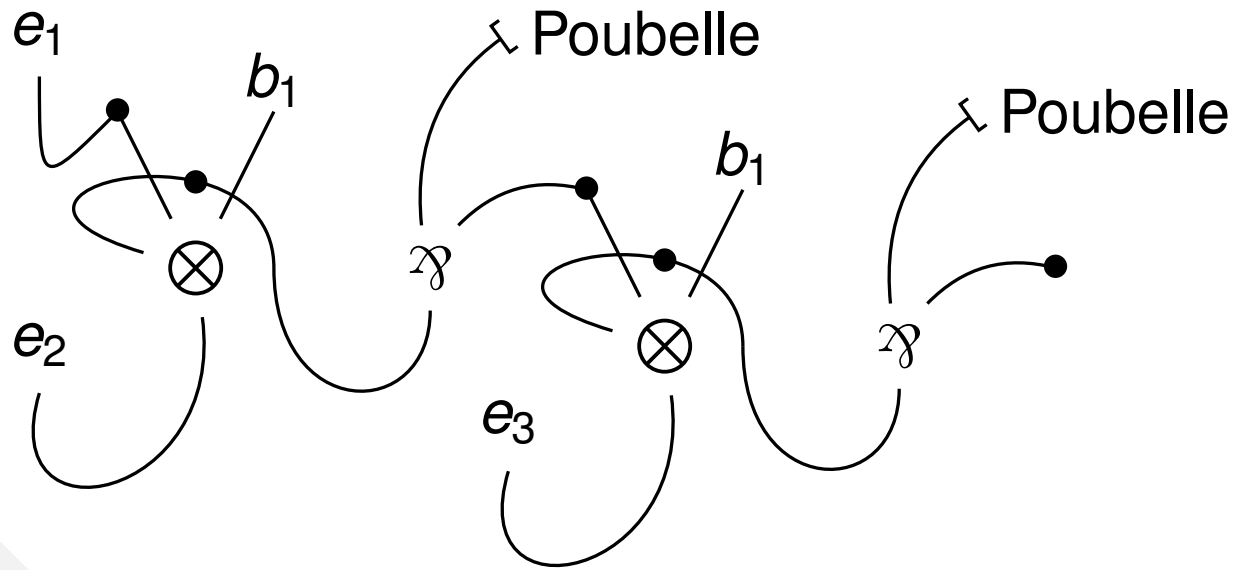


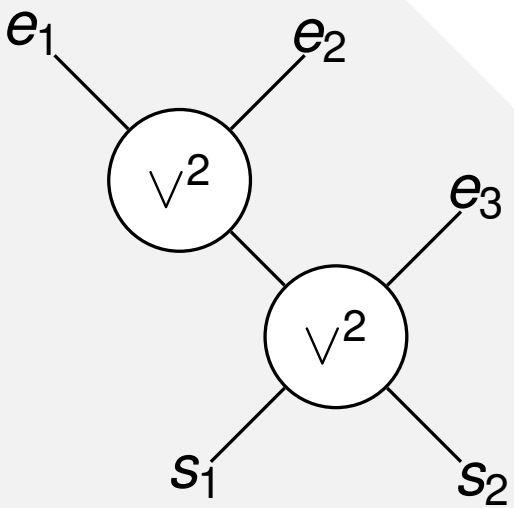
Circuits de preuves : composition



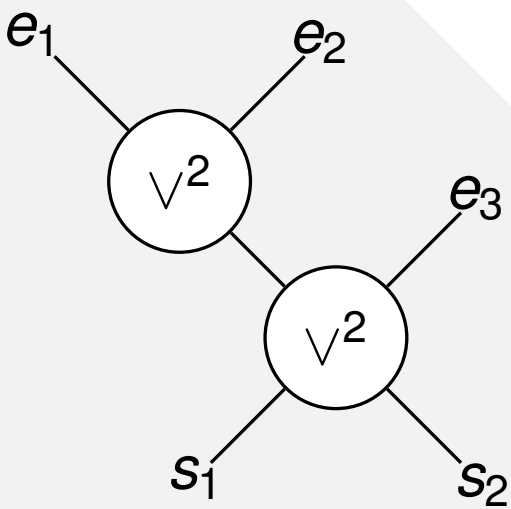


Circuits de preuves : composition





Circuits de preuves : composition



⚠ Difficultés

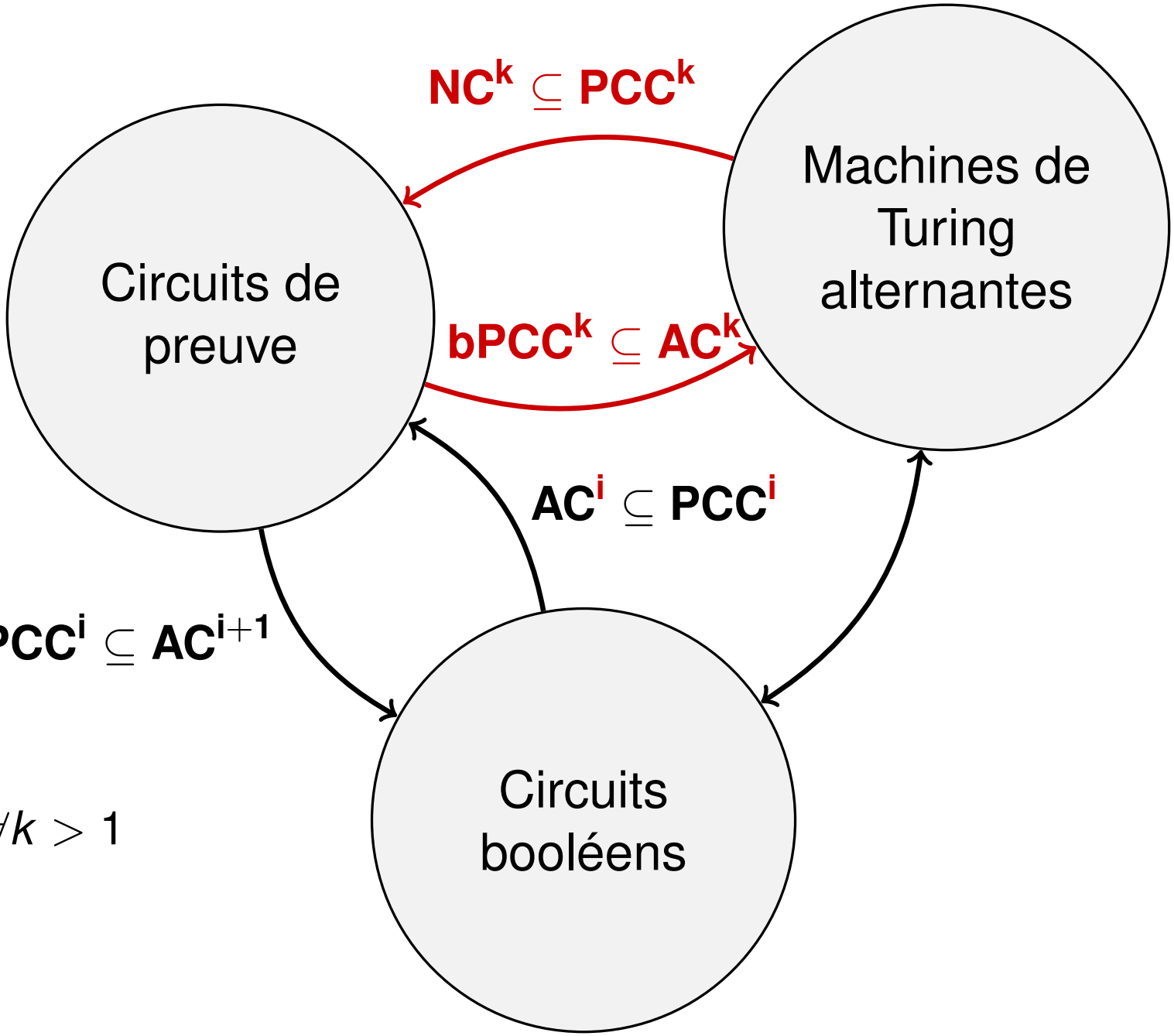
- Langage de description, familles et uniformité
- Évaluation parallèle
- Complexité des réductions

Avancées

Une présentation

- plus modulaire,
- qui simplifie la traduction
- avec un algorithme de normalisation en espace log

Circuits de preuves : Résultats



$\forall i \in \mathbb{N}, \forall k > 1$

Deuxième partie

Approches sémantiques

Objectif

Interpréter dynamiquement les preuves

Correspondance

Preuves	$\leftrightarrow$	Opérateurs
Élimination des coupures	$\leftrightarrow$	Équation de rétroaction
Normalisable	$\leftrightarrow$	Nilpotent

Outils

- Chemins dans un réseau de preuve
- Algèbre d'opérateurs
- Quantiques

Objectif

Calculer avec l'interprétation de preuves dans une algèbre de von Neumann.
Faire passer les restrictions du côté de l'interprétation du langage.

Avancées

- Capturer **L** et **NL**
- Matrices pour représenter le calcul

Calcul et facteur hyperfini : Représenter les entiers

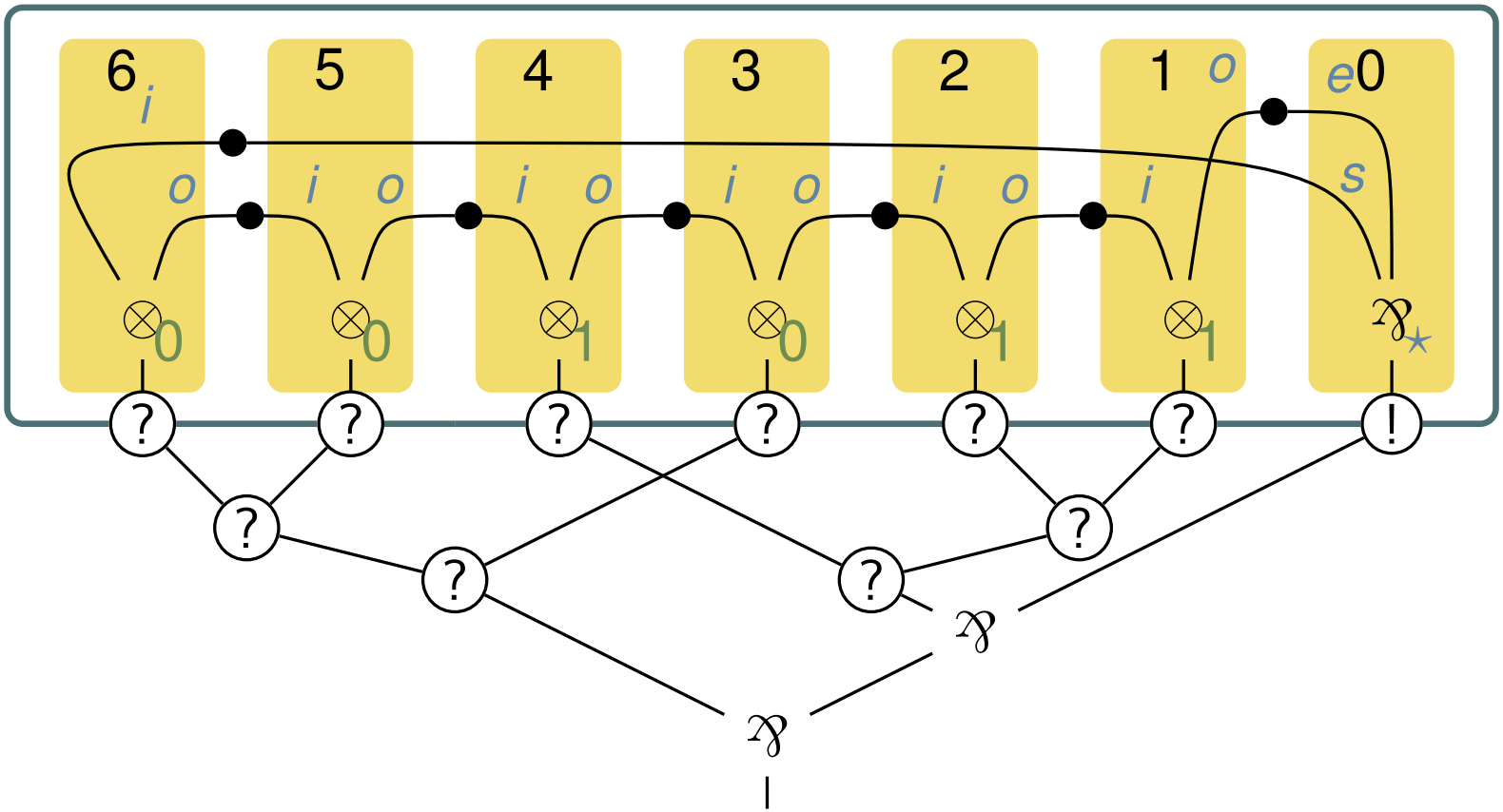
Entiers $\to 0, 1, 2, 3, \ldots$

$\hookrightarrow$ Listes binaires $\to 001, 010, 011, 100, \ldots$

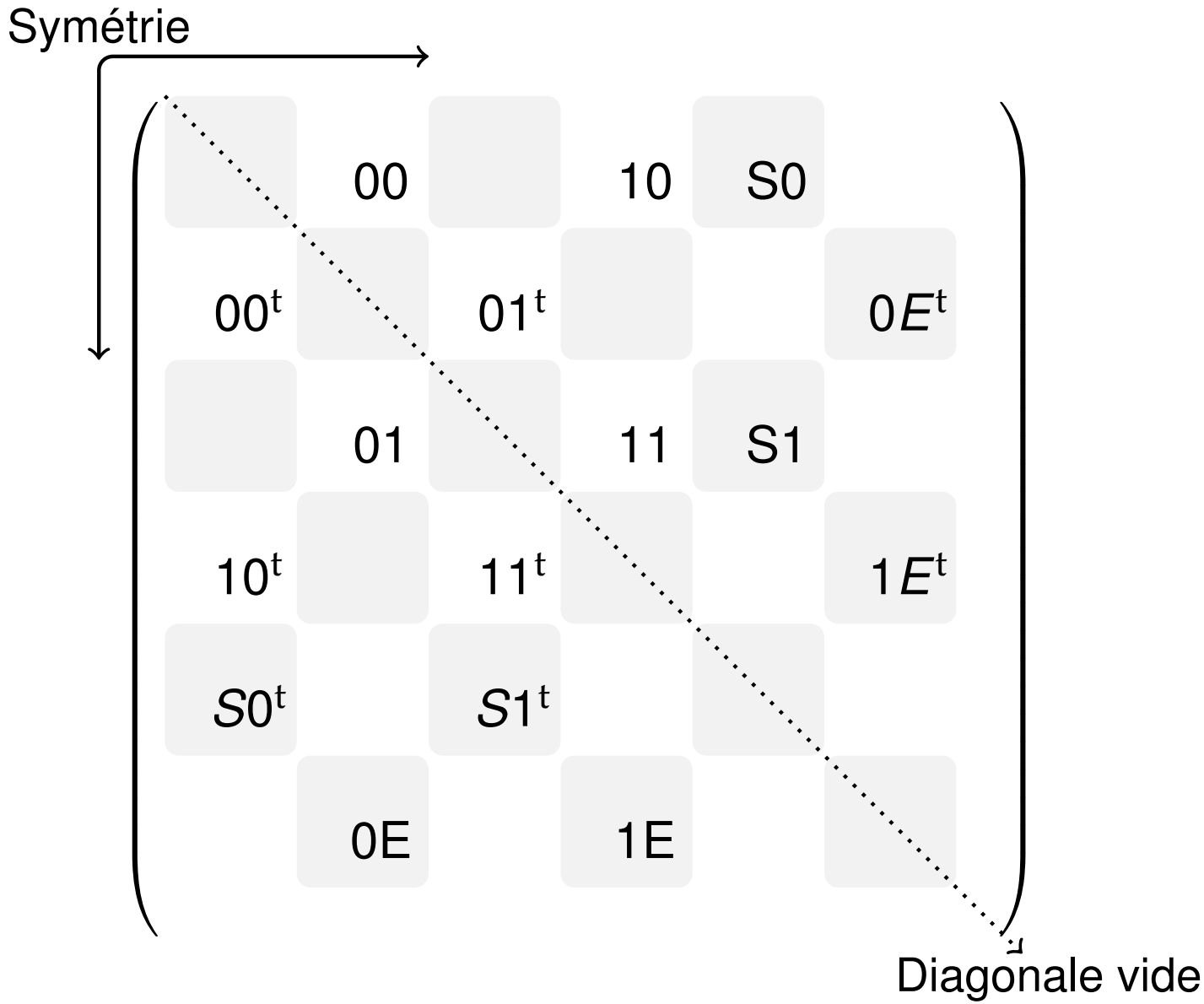
$\hookrightarrow$ λ-termes $\to \lambda f_0 \lambda f_1 \lambda_x \cdot f_0(f_1(f_1(\ldots(f_0 x)\ldots)$

$\hookrightarrow$ Preuves $\to \forall X!(X \multimap X) \multimap (!(X \multimap X) \multimap !(X \multimap X))$

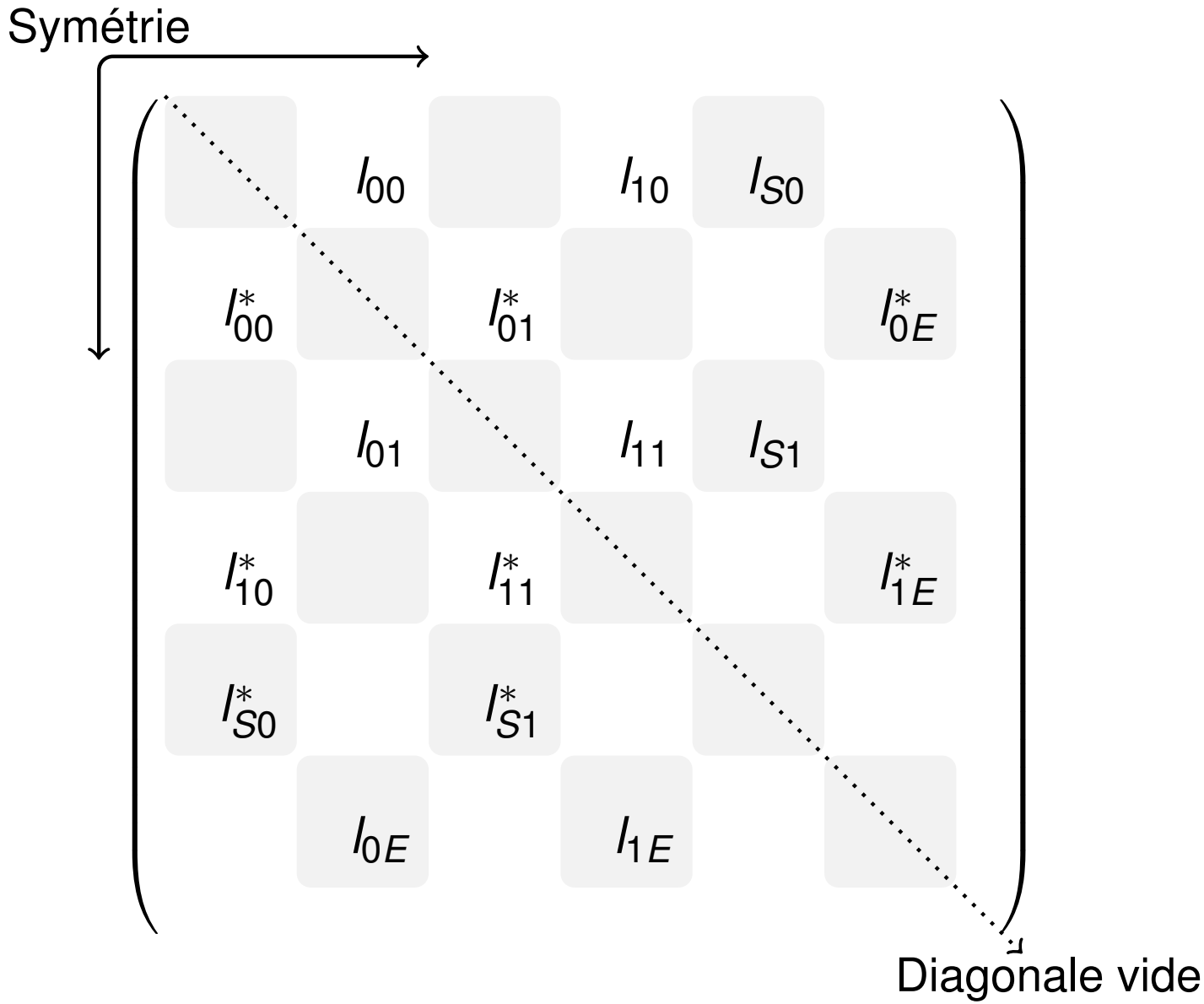
$\hookrightarrow$ Réseaux de preuves $\neg$



Calcul et facteur hyperfini : Matrices et propriétés



Calcul et facteur hyperfini : Opérateurs et propriétés



Définition (Les entiers dans le facteur II_1 hyperfini)

On peut représenter $n \in \mathbb{N}$ comme un opérateur N_n de $\mathrm{M}_6(\mathfrak{N})$.

Définition (Observations)

Une *observation* est un opérateur de $\phi \in \mathrm{M}_6(\mathfrak{S})$.

Définition (Calculer, accepter)

ϕ accepte N_n si $\phi(N_n)$ est *nilpotent*, c.-à-d. si $\exists k \in \mathbb{N}$ tel que

$$(\phi(N_n))^k = 0$$

Définition (P_+ et $P_{+,1}$)

On fixe une paire normative $(\mathfrak{N}, \mathfrak{G})$, pour $\phi \in \mathrm{M}_6(\mathfrak{G})$ une observation et $N_n \in \mathrm{M}_6(\mathfrak{N})$, $[\phi] = \{n \in \mathbb{N} \mid \phi(N_n) \text{ est nilpotent}\}$

$$\begin{aligned} P_+ &= \{\phi \mid \phi \text{ est une observation booléenne}\} \\ P_{+,1} &= \{\phi \mid \phi \in P_+ \text{ et } \|\phi\|_1 \leqslant 1\} \end{aligned}$$

$$\{P\} = \{[\phi] \mid \phi \in P\}$$

Résultats

$$\{P_+\} \subseteq \textbf{co-NL}$$

$$\{P_{+,1}\} \subseteq \textbf{L}$$

⚠ Difficultés

Revenir au cadre fini nécessite un lemme puissant.

Quatrième partie

Machines à pointeurs

Objectif

Établir un modèle qui rende compte du calcul des observations

Avancées

Représenter le calcul

- en développant un modèle *ad-hoc*
- de « bas niveau »
- en questionnant la notion de pointeur

« Pointeurs = log-space » ?

Intuition à cadrer

— Pointeurs comme outil ? JAG, PURPLE,...
Pointeur comme ruban ? SMM, KUM

— Algorithme de REINGOLD ne se résume pas à une manipulation de pointeurs.

Fait classique

Automates finis bidirectionnels capturent naturellement **L**.

Objectif

Adapter ce modèle aux opérateurs.

Machines à pointeurs : Adapter le modèle

La multiplication comme calcul Une « nouvelle copie » de l'entrée et du programme est fournie à chaque étape

↪ Pas de possibilité de modification.

Observation booléennes « Branches » du calcul indépendantes.

↪ Calcul non-déterministe

Nilpotence *Toutes les « branches »* doivent s'annuler

↪ « Universellement non-déterministe »

Produit croisé avec le groupe des permutations Échanger de place plusieurs copies de l'entrée

↪ Plusieurs registres ou pointeurs.

Entrée circulaire, initialisation particulière, etc.

Machines à pointeurs encodées comme opérateurs

⚠ Difficultés

- Trouver une représentation du rejet
- Préserver la contrainte sur la norme
- Décomposition des opérations élémentaires

Théorème

Pour toute machine à pointeur non-déterministe (resp. déterministe) M il existe une observation $\phi_M \in P_+$ (resp. $\phi_M \in P_{+,1}$) tel que pour tout n, $M(n)$ accepte ssi $\phi_M(N_n)$ est nilpotent.

⊸ Résultats

$$\textbf{co-NL} \subseteq \textbf{NPM} \subseteq \{P_+\}$$

$$\textbf{L} \subseteq \textbf{DPM} \subseteq \{P_{+,1}\}$$

Conclusion

Circuits de preuve

Liens entre **AC** et **NC**, **STA**$(\log, *, \log^{\mathbf{i}})$ et **STA**$(\log, \log^{\mathbf{i}}, *)$
Revenir vers le λ-calcul
Non-uniformité (**P**/POLY) : Mazza *Non-Uniform Polytime Computation in the Infinitary Affine Lambda-Calculus* 2014

Opérateurs

Algèbres d'unification
Représenter d'autres données

Machines à pointeurs

Programmer avec des matrices
Machines avec infinité d'états
Restrictions sur automates

Bibliographie sélective I

BAILLOT, Patrick et Kazushige TERUI (2004) « Light types for polynomial time computation in lambda-calculus » dans : *Logic in Computer Science, 2004. Proceedings of the 19th Annual IEEE Symposium on* IEEE, p. 266–275

DAL LAGO, Ugo et Ulrich SCHÖPP (2010) « Type Inference for Sublinear Space Functional Programming » dans : *APLAS* sous la dir. de Kazunori UEDA t. 6461 Lecture Notes in Computer Science Springer, p. 376–391

GIRARD, Jean-Yves (1987) Linear logic dans : *Theoretical Computer Science*, **50**.(1), p. 1–101

———(1998) Light Linear Logic dans : *Inf. Comput.*, **143**.(2), p. 175–204

MAZZA, Damiano (2014) *Non-Uniform Polytime Computation in the Infinitary Affine Lambda-Calculus* rap. tech. L.I.P.N.

REINGOLD, Omer (sept. 2008) Undirected connectivity in log-space dans : *Journal of the ACM*, **55**.(4), 17 :1–17 :24

WADLER, Philip (1991) « Is there a use for linear logic ? » dans : *PEPM* sous la dir. de Charles CONSEL et Olivier DANVY ACM, p. 255–273